[How to use dictionaries in Python](http://www.pythonforbeginners.com/dictionary-data-structure-in-python/how-to-use-dictionaries-in-python/)

13 Oct 2012   | [Dictionary](http://www.pythonforbeginners.com/category/dictionary-data-structure-in-python/)

Tags: [Dictionary](http://www.pythonforbeginners.com/tag/dictionary-data-structure-in-python/)

This post will explain how to use dictionaries in Python.

About dictionaries in Python

Use {} curly brackets to construct the dictionary, and [] square brackets to index it.

Separate the key and value with colons : and with commas , between each pair.

Keys must be quoted

As with lists we can print out the dictionary by printing the reference to it.

A dictionary maps a set of objects (keys) to another set of objects (values).

A Python dictionary is a mapping of unique keys to values.

Dictionaries are mutable, which means they can be changed.

The values that the keys point to can be any Python value.

Dictionaries are unordered, so the order that the keys are added doesn’t necessarily reflect what order they may be reported back.

Create a new dictionary

# In order to construct a dictionary you can start with an empty one.  
>>> mydict={}

# This will create a dictionary, which has an initially six key-value pairs, where iphone\* is the key and years the values

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | released = {          "iphone" : 2007,          "iphone 3G" : 2008,          "iphone 3GS" : 2009,          "iphone 4" : 2010,          "iphone 4S" : 2011,          "iphone 5" : 2012      }  print released |

>>Output

{'iphone 3G': 2008, 'iphone 4S': 2011, 'iphone 3GS': 2009, '

iphone': 2007, 'iphone 5': 2012, 'iphone 4': 2010}

Add a value to the dictionary

You can assign to an individual dictionary entry to add it or modify it

|  |  |
| --- | --- |
| 1  2  3 | #the syntax is: mydict[key] = "value"  released["iphone 5S"] = 2013  print released |

>>Output

{'iphone 5S': 2013, 'iphone 3G': 2008, 'iphone 4S': 2011, 'iphone 3GS': 2009,

'iphone': 2007, 'iphone 5': 2012, 'iphone 4': 2010}

Remove a key and it’s value

You can remove key-value pairs with the del operator

|  |  |
| --- | --- |
| 1  2 | del released["iphone"]  print released |

>>output

{'iphone 3G': 2008, 'iphone 4S': 2011, 'iphone 3GS': 2009, 'iphone 5': 2012,

'iphone 4': 2010}

Check the length

The len() function gives the number of pairs in the dictionary.

|  |  |
| --- | --- |
| 1 | print len(released) |

Test the dictionary

Check if a key exists in a given dictionary by using the in operator like this:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | >>> my\_dict = {'a' : 'one', 'b' : 'two'}  >>> 'a' in my\_dict  True  >>> 'b' in my\_dict  True  >>> 'c' in my\_dict  False |

or like this in a for loop

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | for item in released:      if "iphone 5" in released:          print "Key found"          break      else:          print "No keys found" |

>>output

Key found

Get a value of a specified key

|  |  |
| --- | --- |
| 1 | print released.get("iphone 3G", "none") |

Print all keys with a for loop

|  |  |
| --- | --- |
| 1  2  3  4  5 | print "-" \* 10  print "iphone releases so far: "  print "-" \* 10  for release in released:      print release |

>>output

----------

iphone releases so far:

----------

iphone 3G

iphone 4S

iphone 3GS

iphone

iphone 5

iphone 4

Print all key and values

|  |  |
| --- | --- |
| 1  2 | for key,val in released.items():      print key, "=>", val |

>>output

iphone 3G => 2008

iphone 4S => 2011

iphone 3GS => 2009

iphone => 2007

iphone 5 => 2012

iphone 4 => 2010

Get only the keys from the dictionary

|  |  |
| --- | --- |
| 1  2 | phones = released.keys()  print phones |

# or print them out like this:

|  |  |
| --- | --- |
| 1  2 | print "This dictionary contains these keys: ", " ".join(released)  >>iphone 3G iphone 4S iphone 3GS iphone iphone 5 iphone 4 |

# or like this:

|  |  |
| --- | --- |
| 1  2 | print "This dictionary contains these keys: ", " ", released.keys()  >>['iphone 3G', 'iphone 4S', 'iphone 3GS', 'iphone', 'iphone 5', 'iphone 4'] |

Printing the values

Elements may be referenced via square brackets, using the key:  
print released["iphone"]

|  |  |
| --- | --- |
| 1  2  3  4 | print "Values:\n",  for year in released:      releases= released[year]      print releases |

>>output:

Values:

2008

2011

2009

2007

2012

2010

Printing with pprint

|  |  |
| --- | --- |
| 1 | pprint.pprint(released) |

Sorting the dictionary

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38 | for key, value in sorted(released.items()):      print key, value    <pre>>>output:  ('iphone', 2007)  ('iphone 3G', 2008)  ('iphone 3GS', 2009)  ('iphone 4', 2010)  ('iphone 4S', 2011)  ('iphone 5', 2012)  """  </pre>  <pre class="brush:"python"">for phones in sorted(released, key=len):      print phones, released[phones]  </pre>  <pre>>>output:  iphone 2007  iphone 5 2012  iphone 4 2010  iphone 3G 2008  iphone 4S 2011  iphone 3GS 2009    </pre>  <h3>Counting</h3>  <pre class="brush:"python"">count = {}  for element in released:      count[element] = count.get(element, 0) + 1  print count  </pre>  <pre>>>output:  {'iphone 3G': 1, 'iphone 4S': 1, 'iphone 3GS': 1, 'iphone': 1,  'iphone 5': 1, 'iphone 4': 1}    </pre> |